AD-A071 375 STANFORD RESEARCH INST MENLO PARK CA NAVAL WARFARE R--ETC F/G 15/3
AN APPROACH TO A GAME THEORETIC TREATMENT OF FLEET DEFENSE.(U)
APR 76 W H FRYE N00014-76-C-0633
UNCLASSIFIED NWRC-TN-66 NL

| OF |

AD
A071375

END
DATE
FILMED
8-79
DDC

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

1

Naval Warfare Research Center
Technical Note (14) NWRC-TN-66

(11) 30 April 1976

(6) AN APPROACH TO A GAME THEORETIC TREATMENT OF FLEET DEFENSE,

(9) Technical note,

(12) 73 p.

By: (10) William H. Frye

Prepared for:

NAVAL ANALYSIS PROGRAMS (CODE 431)
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217

CONTRACT (15) N00014-76-C-0633

SRI Project 4840

Reproduction in whole or in part is permitted for any purpose of the United States Government.

DRAFT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

SUMMARY

This technical note shows, largely by means of a fairly complex example, how realistic Fleet Defense problems may be modeled using game theory. Emphasis is placed on the structuring of the tactical problem as opposed to obtaining its game theoretic solution; given the paucity of realistic game solutions in use by the Navy this emphasis is felt to be justified. For the most part, game theory technique per se is of secondary interest at this stage of development--standard two-person, zero sum (matrix) games will often suffice. However, there is one important exception. Standard theory is not readily available for handling uncertainty, and this topic, which is of both practical and theoretical interest is considered at some length. Constrained matrix games are needed to solve games with uncertainty.

An overview of the document follows. The need for a game theoretic treatment of Fleet Defense problems is stated and examples of the kinds of questions that may benefit from an appropriate game theory analysis are given. The overall approach is then outlined; its essential ideas are to:

(1) Use a scenario approach in order to have a definite problem context.

(2) Develop an effectiveness model from the scenario in such a way that the gaming methodology can be readily interfaced with it.

(3) Use a flexible effectiveness measure as a part of this model.

(4) Decompose the overall problem and develop an algorithm for solving the overall game. → next page

(5) Start with perfect information and then introduce imperfect information (uncertainty),

(6) Solve the many lower-order games by whatever methods are appropriate, and

(7) Use the game solutions to study tactical decisions and tradeoffs of all kinds.

A Fleet Defense scenario is then defined, first by a narrative and then by a more precise event-flow diagram. The scenario involves a Carrier Task Force (the Blue player) in transit within range of an enemy (Red player) land base. Red reconnaissance aircraft search for the CV and, upon detection, call in attack aircraft carrying cruise missiles to attack the CV. CAP, DLI, and AEW aircraft are represented and comprise the major elements of the defense. Counterdetection and interception of the reconnaissance aircraft are possible, as are the detection of the raid by AEW and subsequent interception by CAP or DLI. If the reconnaissance aircraft is lost to Red, the raiders are forced to perform their own search for the CV. The last layer of Blue defense is a Point Defense system. Passive/active decisions are an important part of the scenario for both players.

The next step is the crucial one: to translate the event-flow diagram into a form of effectiveness model that is compatible with a game theory solution. The concept is to define a model which, when all decision variables are fixed, is a Markov chain model with absorbing states.

States and state transitions of the model are first defined in such a way that the flow of scenario events is adequately represented, with the transition probabilities being in general functions of the decision variables of the two players. Four kinds of states are distinguished, depending upon the degree of control the two players have over the values of the outgoing transition probabilities. Some states will have transition probabilities independent of both players' decision variables;

these are chance states. Other states will have transition probabilities controllable by just one player, and these define one-sided optimization problems. The more interesting and vital states have transition probabilities under joint control of the two players; these are game states.

Several terminal (absorbing) states are defined in terms of the terminal conditions of the scenario. By associating an input value (actually a utility) with each absorbing state, the effectiveness measure is defined as the average value (or average utility) over a large number of replications of the model. (Put differently, if the Markov chain model were replicated a large number of times in the Monte Carlo manner, a probability distribution over the absorbing states would be determined. This distribution would be used to weight the input utility values to form the effectiveness measure.) In game theoretic terms this measure is defined to be the payoff associated with the decision variables, and the relationship of the payoff to the decision variables is the payoff function. The min-max game is played with this function; Blue wants to maximize this function and Red wants to minimize it.

Dynamic programming has been selected as the natural method for solving the overall game. This method works backwards from what is already known (or evaluated), which in this application means initially working backwards from the absorbing states with their input values. The dynamic programming process "rolls back" from evaluated states to unevaluated states. At each state a mathematical problem whose type depends on the type of state is presented. At game states a two-person, zero-sum game is solved, in the absence of uncertainty an ordinary matrix game will often suffice. A detailed example of a model at a game state is given.

The last part of the document is devoted to uncertainty. A model is given which uses the concept of a lottery; the game to be played is

chosen by a lottery whose probabilistic functioning both players know. The text has an intuitive argument to justify the assumption of this kind of uncertainty model. By this device a game with uncertainty can be handled as a game with perfect information, but the information now has to do with the probability distribution behind the lottery and not tactical information itself. The detailed game-state model mentioned above is reformulated with uncertainty, and a constrained matrix game is given for its solution. Linear programming may be used for the numerical solution of this lottery game.

There are two appendices: Appendix A further develops the geometry of the Markov model, and Appendix B elaborates on the uncertainty model and gives a numerical example.

CONTENTS

V A DETAILED MODEL AT A GAME NODE (Continued)

APPENDICES

ILLUSTRATIONS

TABLES

I INTRODUCTION

This technical note contains results obtained in the first phase of a project whose overall objective is the application of game theory methods to realistic Fleet Defense problems. The document is primarily concerned with the difficult and fundamental problems of structuring a complex problem so that game theory methods can be used to obtain solutions at a reasonable expenditure of time and effort.

There are at least three reasons for applying game theory to Fleet Defense problems. First, analytical solutions are badly needed to help understand and resolve the maze of tradeoffs that are involved in Fleet Defense at the tactical level. These complex problems have been found tractable only by Monte Carlo simulation. Such methods have their merits but they do not provide a theoretically justifiable framework for resolving the many tactical choices and tactical and equipment tradeoffs which exist. Optimization methods generally and game theory in particular can, when properly used, provide structure to complex, hard-to-define real problems. Second, analytical methods used on this class of problems are usually one-sided--the opponent's tactics and strategy being established by assumption and not analysis. Any two-sided analysis which has been done tends to be ad hoc and the methods are not readily transferable to other problems. Finally, game theory makes it possible to disentangle the contributions of equipment and tactics. Specifically, because the tactics are optimized, any comparison of effectiveness derived for differing equipment (or equipment parameters) will more truly reflect the differences in value of the equipment itself.

?

This research effort has an ambitious goal: to bring ordinary zero-sum, two-person game theory to bear on practical Fleet Defense problems in such a way as to be of interest and use to those who make Navy tactical decisions and to the naval analysis community.

Some of the kinds of questions which can be examined using the methodology to be discussed are:

- Whether to have CAP aircraft or not, and if so, how many and where?
- How should available AEW aircraft be used?
- How should ships of a task force be deployed? (Close formation vs. dispersed formation questions.)
- Hard kill vs. soft kill questions.
- When should EMCON be used, and what should be the conditions for breaking it?
- How should CAP surface-to-air missile coordination problems be resolved?
- In a multiple-threat environment, how should the defense balance its forces?

The primary outputs of games solved using the methodology are the optimal values of the decision variables for the scenario. Many of these are probabilities, e.g., the probability that CAP should be used, the probability that AEW aircraft should be used, and the probability that the CV should be initially in EMCON. As special cases, these probabilities are often expected to be zero or one. When this is the case, the decision variables are interpreted as whether or not to use CAP, to use AEW, or to be initially in EMCON.

Not all decision variables are probabilities, however. For example, in a more advanced model than presented here the range and bearing of a CAP station would be decision variables and their optimal values would indicate where CAP should be positioned.

Because the effectiveness model is designed to stand on its own when all decision variables are fixed, one also has the effectiveness model at the optimal point to exercise. Many of the usual things that are done with effectiveness models can be done with this model also, providing care is taken to interpret optimality. To illustrate this, we point out that sensitivity analysis performed at the optimal point by varying some parameter which is not a decision variable will in general be nonoptimal. Such sensitivity results may be adequate for many purposes, however.

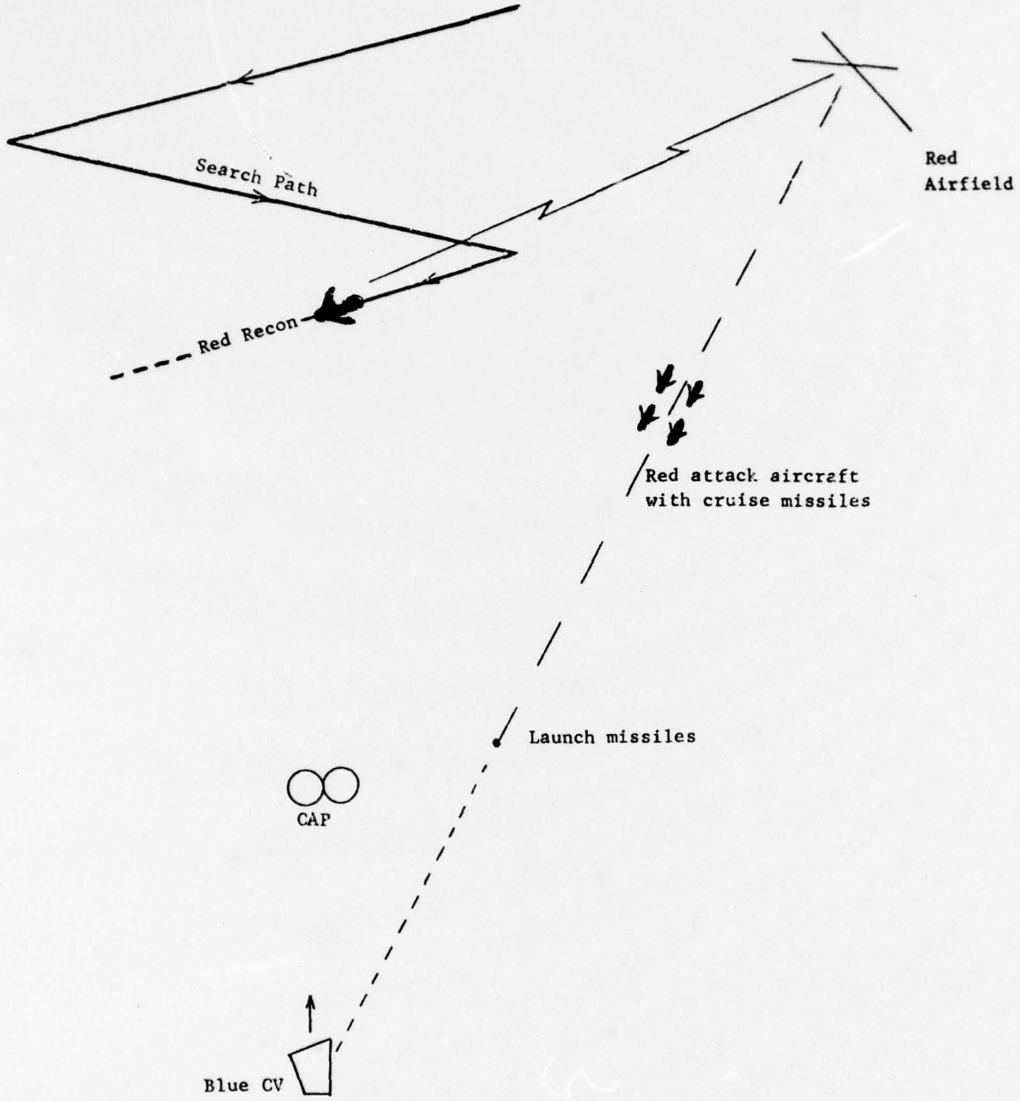
II A FLEET DEFENSE SCENARIO

This section gives a verbal sketch of the fleet defense problem to be modeled and solved using game theory. Although its structure is fairly simple, it contains a number of the essential aspects of the real tactical problem. The problem is more fully defined and occasionally simplified in later sections. Following gaming tradition we label one player (or side) Blue and the other Red; we arbitrarily choose Blue to have the Fleet Defense problem and Red the search and attack problem.

Consider a Blue Carrier Task Force (CTF) in transit, passing within range of a Red land base capable of supporting attack aircraft (see Figure 1 for the geometry). Blue wants to complete the transit without loss or damage, while Red wants to sink or at least damage the CV. (The larger scenario in which this problem is imbedded must be translated into the payoff structure of this model by means to be explained later.)

The CV may choose to deploy Combat Air Patrol (CAP) aircraft for search, investigation, and intercept. (By CAP is meant a combination of fighter and Airborne Early Warning (AEW) aircraft.) The AEW aircraft may search either actively or passively. The CV has several important tactical choices to make at the outset, one is whether to be in EMCON or to be active. If initially in EMCON the CV has opportunities at later critical times to "go active," which means essentially that the CV turns its radars on.

A Red reconnaissance aircraft ("Recon") flies along the path shown in the figure, searching for the Blue force. Recon may detect either the CV or one of the CAP aircraft, and, upon classifying the CV to a



Notes: i) Recon detects CV or CAP, calls in attack aircraft.
2) If TF detects Recon, may try to intercept before detection of CV or launch of cruise missiles.

FIGURE 1 A SIMPLE FLEET DEFENSE PROBLEM

sufficient level of confidence, calls in the attack aircraft for an attack on the CV. (Search by Recon can be in either the active or passive mode.)

The attack aircraft then begin flying out towards the CV, using position information provided by Recon. Since some time is required for the flyout, and since in any case the position information is imperfect, the attack aircraft need further assistance in locating the CV. Knowing this, the Blue force may attempt to destroy the Recon before the attack aircraft launch their weapons (cruise missiles) at the CV. (For the geometry shown the interception would probably be performed by CAP fighters. Recon has orders which govern his behavior in the event of attack by Blue--he may either flee if the situation warrants it or be forced to stay.)

AEW aircraft are in the meantime searching, and may detect the incoming raid even at low altitude. Detection may permit interception by CAP or Deck Launched Interceptors (DLI) to counter the raid.

If the Recon aircraft should be shot down (or driven from the scene) while the aircraft in the raid still need it as an information source, one or more raiders may choose to "pop-up" from their assumed low altitude profile to search for the CV. (The raid will be flying at low altitude in the last phase of the run-in in order to avoid detection by the CV's radars.) The pop-up may or may not result in detection, when detection is not obtained the raid has no choice but to return home empty handed. If detection is obtained during the pop-up, the raiders are assumed to have enough CV information to be able to launch their cruise missiles.

The pop-up is not without its dangers for Red, because Blue may detect this brief maneuver and thereby prepare its point defenses against the incoming missiles themselves. The point defense may or may not be effective, given an opportunity to bring it into play, and the number of hits suffered by the CV will vary accordingly.

From this sketch, several distinguishably different outcomes may be listed:

(1) There is no detection by either side in a reasonable length of time

(2) The Recon is lost and the attack aircraft return home empty handed

(3) The attack aircraft are shot down before launch of their missiles

(4) The missiles are shot down by the CV's point defense system

(5) Missiles impact the CV, resulting in damage or sinking.

The payoff structure will later be constructed using these terminal conditions.

A pictorial form of the scenario is shown in Figure 2, it is based on the critical events of the scenario. The diagram may be considered an event-flow diagram or precedence relation diagram. It should not be considered a state diagram for technical reasons to be discussed later. The diagram consists of nodes and arcs and is technically a directed graph. The double-ellipses denote terminal nodes.

A convenient way to relate the diagram and the scenario is to consider sample paths (i.e., sequences of nodes) from the starting node to a terminal node as possible "plays" of the scenario. The simplest possible play is path (1-12), the path from starting node 1 to terminal node 12, representing the situation in which neither side detects the other. Probably the next simplest (in terms of the amount of interaction between the Blue and Red forces) is the (1-2-9-15) path. This path may be explained as follows: Red detects Blue at a random time T_2, but Blue either does not detect Red ($T_1 = \infty$ or T_1 is large relative to T_2).

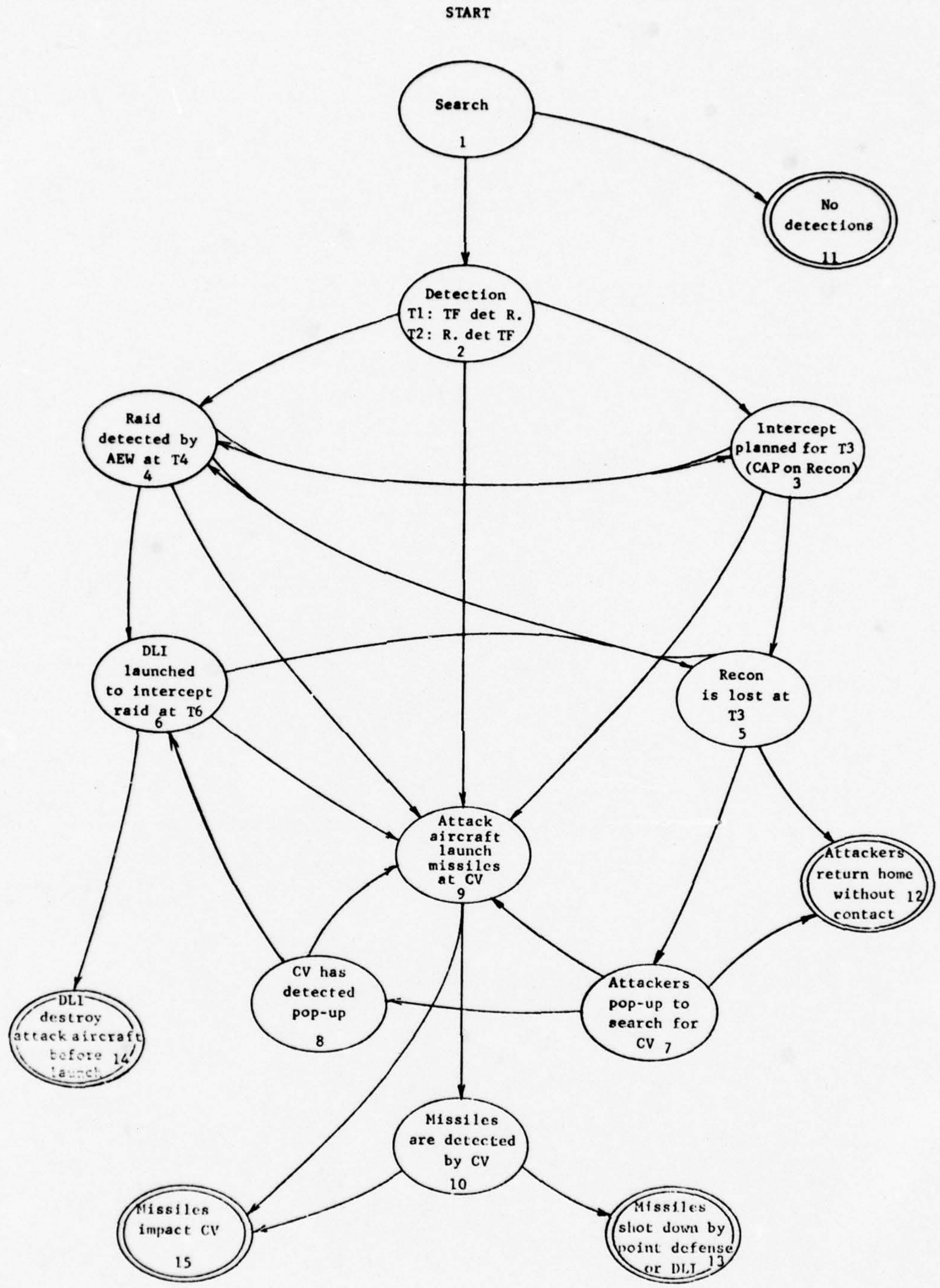

START
Search
1
No detections
11
Detection
T1: TF det R.
T2: R. det TF
2
Raid detected by AEW at T4
4
Intercept planned for T3 (CAP on Recon)
3
DLI launched to intercept raid at T6
6
Recon is lost at T3
5
Attack aircraft launch missiles at CV
9
Attackers return home without contact
12
CV has detected pop-up
8
Attackers pop-up to search for CV
7
DLI destroy attack aircraft before launch
14
Missiles are detected by CV
10
Missiles impact CV
15
Missiles shot down by point defense or DLI
13

FIGURE 2 EVENT FLOW DIAGRAM FOR THE SCENARIO

Attack aircraft are called in by Recon, the raiders run out from their airbase unopposed and launch cruise missiles at the CV at time T_5. The missiles are not detected by the CV, and therefore impact it, ending the play of the game.

A more complex set of interactions is given by the path (1-2-3-4-5-7-9-10-13). Using hypothetical values for the several time instants involved, a typical play may be described as below. All times are in minutes, measured from an arbitrary reference.

<u>Node</u> <u>Explanation</u>

1 Search. Both sides search according to their selected search mode.

2 Detection: The CTF detects Recon at $T_1 = 35$
Recon detects the CTF at $T_2 = 40$
and the raiders are called in.

3 An intercept is planned: CAP will intercept the Recon at $T_3 = 42$.

4 An AEW aircraft detects the raid at $T_4 = 55$ (7 minutes before the raid plans to launch its cruise missiles at the CV).

5 Recon is lost at $T_3 = 42$, which forces the raid to either pop-up to search or to return home.

7 One of the raiders pops up at $T = 60$. The CV is detected and the planned missile launch time is $T_5 = 62$.

9 The CV does not detect the pop-up, so the aircraft launch their missiles at $T_5 = 62$.

10 The missiles are detected by the CV (in spite of the CV's missing the pop-up).

13 The missiles are shot down by the CV's Point Defense.

III MODELING APPROACH

This research effort has been devoted to finding gaming methods of use in analyzing Fleet Defense problems. Game theory requires a payoff structure, and payoff in a zero-sum context can be satisfactorily defined in terms of an effectiveness measure. Thus a necessary step in the development of a gaming methodology is the development of an effectiveness* model. This can, of course, be a large problem in itself.

A. The Effectiveness Model

The approach that has been taken is to first design an effectiveness model to meet two basic requirements:

i) The model should be an adequate effectiveness model in its own right

ii) The model should be structured so that the gaming model can be readily defined in terms of the effectiveness model elements.

The thrust of the first requirement is that the effectiveness model should have as parameters all of the decision variables of both players, and when these parameters are fixed the model should "run" as an effectiveness model, giving an outcome or a probability distribution of outcomes from which effectiveness is or can be determined. By the second requirement we imply that the game model will be compatible with the effectiveness model so that the two will form an integrated whole. A design that will not be used is one in which the effectiveness model merely provides, in a straightforward manner, elements for the payoff

*Payoff and effectiveness are essentially synonymous in this document.

matrix of a game which is then routinely solved separately. Specifically, we are not considering an OPSTA I approach.

1. Markov Model

With this general approach established, preliminary experimentation with various kinds of Fleet Defense situations showed the desirability of having some general, overall model structure to work with. Upon briefly surveying the possibilities and considering the various features of different model types, it was decided that some form of Markov model should be attempted. Elsewhere, such models have been found to be very flexible and adaptable to many tactical situations, and mathematically they have many qualities of linearity which may be helpful. Furthermore, there is widespread knowledge of Markov models and their properties, and definitions are both well known and standardized.

Why?

probably because of The desire to Have A model WITH Reasonable prospects of obtaining A Solution,

Should the be The criteri for selectio of model

Figure 2, the event diagram shown earlier, can already be considered as the beginning of a Markov model by considering a node to be a state. The terminal nodes introduced for payoff purposes can be regarded as absorbing states. Other nodes are transient states, and a play of the game corresponds to a Monte Carlo replication of a Markov model from the starting state to absorption. The absorption probabilities are needed for an effectiveness measure. For each terminal state t the probability of absorption p_t in that state can be calculated. If V_t is the payoff specified as input for terminal state t, the average payoff $\overline{V}$ is defined to be:

$$\overline{V} = \sum_t p_t \cdot V_t \quad .$$

Further experimentation led to the concept of the gaming model presented in this technical note: some form of Markov model would first be defined as above, consisting of states and transition probabilities.

Transition probabilities would then be worked out as a function of the decision variables and the other parameters of the problem. The decision variables would be used to play the game after suitable decomposition. This form of model is a version of what is technically known as a Markov Stochastic Game*.

2. Decision Variables

In this document we will seldom use the conventional game theory terms strategy, pure strategy, and mixed strategy. We prefer to use the term decision variable to include them all; a decision variable for Blue (Red) is a variable controllable by Blue (Red), often subject to constraints. In a constrained matrix game context experience has shown that descriptions are more natural using "decision variable" than those made in terms of strategies.

An example which relates the two definitions is in an ordinary matrix game: the decision variables are the probabilities of selecting the pure strategies. These variables are constrained to be non-negative and to sum to unity for each side.

There are four types of states classified by controllability of the transition probabilities on outgoing arcs. Transition probabilities out of some states may be controlled only by Blue, and those out of other states controlled only by Red. These two types of state will represent one-sided optimization problems and can be called "red-controlled states" or "Blue-controlled states." The more vital states are those whose transition probabilities are under joint control--these are called "game states" since a game will be played at each of them. A fourth type of state has transition probabilities not controllable by either Red or Blue, these are "chance" states.

*The payoff structure in such games is richer than that required here--a payoff can also be specified for each i-to-j transition.

The idea then is that Blue should choose his decision variables in such a way that a play of the game will tend to end in a terminal state with a relatively large payoff. Red, on the other hand, should choose his decision variables in such a way that the opposite happens--termination tends to occur in states with lower values. In all of this it is understood that sufficient mixing (randomization) of strategies is allowed to give a true min-max solution in the basic game-theoretic sense.

3. The State Space

So far we have the concept of an effectiveness model whose dynamics derive from the possible paths through an event-flow diagram and whose measure is defined in terms of the terminal nodes of this diagram. The model dynamics need to be considered in some detail, and this entails basic considerations about constructing a state space. The usual Cartesian product method of defining states (choosing all combinations of all variables) results in a state space which grows large multiplicatively. Many or even most of the states defined this way are either meaningless or very unlikely to be used. This is unsatisfactory in the present context; a better way is needed.

In the present context, a state is essentially information which is sufficient to define a game problem. If Figure 2 were a state diagram in which nodes were states, it would be possible to completely define a game at each node. However, by working through a few situations using Figures 1 and 2, one finds that the node labels are insufficient to define state. More information is needed and the question is how to systematically and conveniently provide it.

One convenient way to add information is to consider not just where the flow is but how it got there, i.e., to consider the path to the node as well as the node itself. In terms of the elements of Figure 2, a state could be defined to be a path from the starting node (search) to some other node, together with any instants of time T_i encountered along this path. Because of restrictions inherent in the dynamic programming methodology which will be used to solve the game, it would also be necessary to restrict the paths to those without loops. That is, no path could include the same node more than once.

4. Aggregation

This method of defining state is quite systematic and easy to work with. However, in this example, only about one fourth to one half of the states would actually be needed because some or even all of the early path data becomes irrelevant as the game proceeds. To reduce the number of states one can aggregate two or more paths which are indistinguishable from the game standpoint and consider the result an aggregated state.

As an example of aggregation, consider the states passing through node 9 labelled "Attack Aircraft Launch Missiles at T_5." Upon arrival in this node, the launch occurs and Blue's problem is to detect and shoot down the missiles. Quite a number of events which occurred earlier are now irrelevant, in particular it does not matter whether node 5 (Intercept Planned) and/or node 8 (Attackers pop-up) were passed through or not. Thus paths (1-2-3-9) and (1-2-3-5-7-8-9) and (1-2-3-5-7-9) can be aggregated, i.e., replaced by a path denoted (1-2-3-x-10), say. Time instant (T_i) information would be aggregated or eliminated in an obvious way and the result associated with the aggregated path to form a single aggregated state to replace the orginal states.

There will be little mention of path or state aggregation in the sequel, it suffices to say that systematic, programmable methods are being considered. Unless otherwise indicated, for convenience in the rest of this document we will always consider a state to be the complete path and the associated times. Appendix A provides more detail on the geometry implicit in Figure 2, showing a detailed tree diagram and using it to determine the number of states and the distribution of path lengths.

5. An Example

A simple hypothetical example will illustrate much of what has been discussed, and will furthermore serve to introduce some of the mathematical ideas to come. Figure 3 shows a Markov effectiveness model example with three transient and three absorbing states. Because the transition probabilities are dependent only on the node and not on the path we may equate node and state, i.e., aggregate all paths to a node into a state, and identify the node with the state. Payoffs (to Blue, who maximizes) are 3, -2, and 10 for terminating in states 4, 5, and 6, respectively, as shown on the diagram. Each transient state can make transitions to three other states as indicated by the directed arcs. Algebraic formulas are given on the arcs for transition probabilities; for example, from state 1 (the starting state) the probability of making a transition to state 4 is $2*x_4$, to state 3 the transition probability is y_4, and to state 2 it is $(1-2*x_4 - y_4)$. The x_j and the y_i are the decision variables for Blue and Red, respectively.

By summing the transition probabilities out of each state one sees that they add to one, as required. However, further constraints have to be added to make the probabilities each lie in the interval zero to unity. Constraints selected for this purpose are:

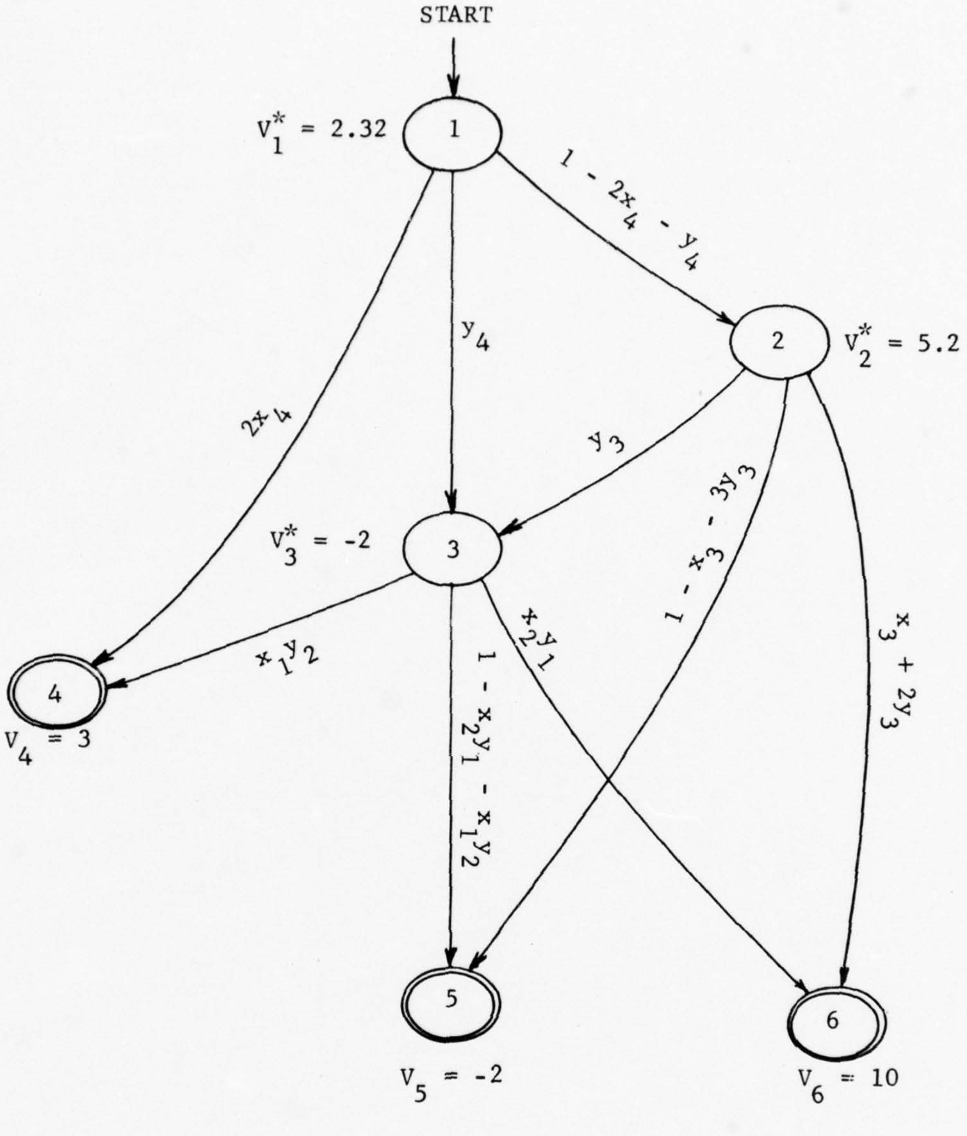


FIGURE 3 MARKOV MODEL FOR A SIMPLE GAME
(Blue's decision variables are x_1, x_2, x_3, x_4)
(Red's decision variables are y_1, y_2, y_3, y_4)

$$.2 \le x_1,\ x_2 \le .6$$

$$0 \le y_1,\ y_2 \le .8$$

$$.1 \le y_3 \le .3$$

$$0 \le x_3 \le .4$$

$$0 \le x_4 \le .3$$

$$0 \le y_4 \le .4 \quad .$$

With these constraints we have a well-defined game with a min-max solution in the decision variables x_1 - x_4 and y_1 - y_4.

Solution of the entire game without decomposition appears to be quite difficult but solution by dynamic programming is fairly direct. We proceed by working backwards from states whose value is already determined, and at this point only the terminal states have their values determined (they are the payoffs 3, -2, and 10). Since the diagram is loopless there has to be at least one other state which can transition only to evaluated states, here it is state 3. Thus we set up a constrained matrix game at state 3 in which Blue selects x_1 and x_2 and Red selects y_1 and y_2 subject to their constraints. Letting * denote "optimal," the solution to this game is $x_1^* = x_2^* = .2$, $y_1^* = y_2^* = 0$, and the value of the game is $V_3^* = -2$. (This value is the mean payoff in a game started from state 3, when both players play optimally from state 3 until the end of the game.) State 3 is now considered evaluated, and the value V_3^* may be associated with state 3 making it, in effect, a terminal state. A second game is played at state 2; Blue chooses $x_3^* = .4$ and Red chooses $y_3^* = .1$ for optimal play, with value $V_2^* = 5.2$. Thus if state 1 were removed from the problem entirely and the game played with state 2 as starting state, the value of the game would be 5.2. The state 2 game, it should be noted, could not have been played before the state 3 game because the value at node 3 was unknown.

The final step is the solution of the game at state 1, which is played in the same manner. For optimal play $x_4^* = 0$ and $y_4^* = .4$, with value $V_1^* = 2.32$. Since state 1 is the actual starting state, the value of the entire game is 2.32 overall, the probabilities of absorption into states (4, 5, and 6) are (0, .64, .36). The saddle point property states that Red cannot reduce the mean payoff below 2.32 if Blue uses the optimal x_i^* listed above for each game, a similar statement holds for Blue when Red plays optimally.

B. The Game Model

The previous section discussed the form of the Markov effectiveness model, with states tentatively defined as loopless paths to nodes and with absorbing states designed so that a satisfactory effectiveness measure could be defined. Transition probabilities are often functions of the decision variables of both Blue and Red, although they may on occasion be functions of Red's variables only, or Blue's variables only, or even be independent of both Red's and Blue's variables. As important as it is, the effectiveness model is auxiliary to the gaming model to be discussed next.

The fundamental elements of this Markov model (like those of any Markov model) are the states, and it has been pointed out that games are theoretically played at the state level. However, the structure of this problem will permit us to play games at the nodal level instead. Thus, a single game can be defined at a node to represent the entire family of states at the node, with the parameters of these states regarded as parameters of the nodal game. For purposes of clarity, however, the discussion will assume that games are played at states.

1. Dynamic Programming

The overall game has already been described: Blue wants to maximize the average payoff and Red wants to minimize it, where the average payoff is the weighted average of the input utility values at the absorbing states. This game will generally be far too large to solve as a unit, however. Therefore decomposition of some kind is required, and we have chosen a Markov chain model for this purpose. Subgames can then be played at the level of a state and their solutions linked together by dynamic programming to solve the overall game*.

A finite Markov model without loops is an almost ideal candidate for solution by dynamic programming. The central idea in the dynamic programming approach, working backwards from already evaluated states, has already been shown by example. The goal is to find, for each state, the game value at that state and the optimal strategies at that state. The game values will "propagate" through the solution because the solution at a state depends upon the game values at already evaluated states. Optimal decision variables do not propagate in this way. Therefore, to form an uncluttered geometrical image of the dynamic programming process, one can visualize recording the game values, as they are found recursively, near the state symbol on the state diagram. This recording serves also to indicate that this state has been "evaluated." Absorbing states, with their values specified as inputs, are by definition "already evaluated." "Working backwards" is not usually uniquely defined, since there is usually more than one way to choose the ordering of states. To start the process, one scans the

*Other methods are probably available as well, in general they use either value iteration or policy iteration methods. Here we discuss only the dynamic programming approach, others may be found in Pollatschek and Avi-Itzhak's Algorithms For Stochastic Games With Geometrical Interpretation, Management Science, Vol. 15, No. 7, (March 1969).

state diagram, looking for some state which can transition only to already-evaluated (i.e., absorbing) states. Because the state diagram is loopless such a state can always be found. Suitable calculations are made at that state according to its type. If it is a game state, the game will be solved by whatever method is required; ordinary matrix games and constrained matrix games are expected to suffice most instances. Both of these types of games are soluble by Linear Programs. If the state has its outgoing transition probabilities controllable by only one player, a one-sided optimization is performed and the resulting value recorded as the game value. If neither player has control of the transition probabilities, the calculation reduces to a straightforward probability calculation without optimization. In any case this state joins the "already evaluated" list and the search begins for another state which has transitions only to already evaluated states. The process continues until all states have been evaluated.

Somewhat more formally, let x_s and y_s be the decision vectors for Blue and Red at state s, and let $p_{s,e}(x_s,y_s)$ be the transition probabilities from s to evaluated states e. Let $\bar{V}_e$ be the value at state e which has already been found and recorded. At node s the payoff function is

$$V_s(x_s,y_s) = \sum_e p_{s,e}(x,y) \cdot \bar{V}_e$$

and the game is to min-max this expression. The result is a pair of optimal decision vectors x_s^* and y_s^*, and a game value V_s^*. A constrained matrix game is used whenever x_s and/or y_s must satisfy additional linear constraints $Ax_s = a$ and/or $By_s = b$.

2. Extensions

This relatively simple structure will be refined and extended when applied to the Fleet Defense model. Briefly, there are two major modifications needed:

- Iteration of the entire process will be required due to the way uncertainty will be modeled.
- Decision variables which occur at more than one state will probably have to be introduced intc state itself. (Decision variables may not be associated with nodes in the convenient way they were in the example. One would like to have decision variables distributed as they were in the example--i.e., occurring at a unique state. When this is the case, a decision variable may be evaluated as soon as it is encountered in the dynamic programming algorithm; when this is not the case, a multiple solution has to be carried along.)

Both of these items are discussed further in later examples.

The impact of these modifications is, as always, on model size and/or running time. Iteration only influences run time, total time will be roughly proportional to the number of iterations. Carrying along multiple solutions, on the other hand, will influence both run time and model size. Fortunately, neither factor is expected to limit the feasibility of the overall approach.

IV FLEET DEFENSE MODEL

We now return to the primary purpose of this research and begin to further develop a Fleet Defense Model. The event-flow diagram presented earlier as Figure 2 is the basis for the real world problem and the previous section is the basis for the methodology.

A. A Partial Summary

A summary will be made at this point to unify the operational elements and mathematical ideas comprising the model. Figure 1 shows the geometry of the scenario and involves the CV, Recon, CAP, airfield, attack aircraft, and cruise missiles launched by the attack aircraft. Figure 2 has nodes representing events or conditions, and a path may be traced from the starting node (search) to any of five terminal nodes, to represent one play of the game or one realization of the scenario. A utility (or payoff) is associated with each terminal node; Blue's objective is to maximize the average value of utility (or payoff) and Red's objective is to minimize it; this is where the zero-sum assumption is made.

A state is tentatively defined to be a path to a node, i.e., it is an ordered sequence of nodes. Any timing information picked up along the path is carried along with it to complete the state definition. Thus, 1-2-3-5 and (T_1 = 30, T_2 = infinity, T_3 = 40) is a state; the corresponding problem defined for that state is to maximize payoff given that the Task Force detected Recon at T_1 = 30, the Recon did not detect the Task Force, and the Recon was lost at T_3 = 40. Present time is considered to be T = T_3, since by convention the node label applies at the instant the node is entered.

In terms of its states, the primary parameters of the model are transition probabilities, which depend in general upon the decision vectors of both Red and Blue, the state, and the T_i associated with the state.

For any given values of the decision vectors x and y , the probabilities of absorption in each terminal state can be calculated, and from these the average payoff determined. Blue chooses x in order to influence the flow to terminate (on the average) in a state with relatively high payoff, while Red chooses y in order to influence the flow to terminate in states with low payoff. Components of the decision vectors may be physical quantities and/or probabilities, in particular some of them may be the probabilities of selecting from the defined decisions at the state.

how?

B. General Observations

Several general observations should be made about the event-flow diagram from which the Markov model is defined. First, it is described as though there were just one unit of each type (one AEW aircraft, one CAP interceptor, etc.). This has been done for simplicity, multiple units can be added later. Similarly, various outcomes are considered to have only two possibilities: the Recon is lost or it is not, the attack aircraft detect the CV or they do not, and missiles either impact the CV or the missiles are shot down by point defense. It should be fairly obvious that more refined outcomes can be added once the basic structure is finished. Another way in which outcomes can be refined is to use the "path memory" that a state may have. That is, because the absorbing states may remember the paths taken to reach them, the utilities V_i specified as inputs may be made a function of path as well as terminal node. As a special case of this a utility term may be defined at an important event at an interior (transient) node. This payoff is analogous to the stagewise payoff $h_1(x,y)$ in N-stage games analysis.

The named times (T_1 through T_6) are needed for the model's logic and decisions based on them are similarly treated in a black-white manner. If, for example, the Recon is lost an instant before the attack aircraft are due to launch (i.e., if $T_3 < T_5$), then the attacker is forced to either pop-up to search or return home without contact. These somewhat unrealistic features can be improved at some cost in terms of complexity at a later time.

Another important thing to point out about Figure 2 is that it is described from the "true" point of view. Equivalently, we might say that at this point the game is played with perfect information for both players. In particular, each side knows whether it has been detected and what time the detection occurred. In a later section when uncertainty is considered, the perfect information assumption will be removed. However, the flow of the model should still be considered to be flow of the true situation, and not the estimate that Blue or Red has of the situation. Uncertainty will be handled on a node-by-node basis.

Some remarks about timing are also needed. We wish to have the effectiveness model structured so that transitions in the model occur in a time sequence, that is, if a path goes from node A to node B the time of entry of A should ~~not~~ be earlier than the time of entry of B. With the exception of node 2 this holds true on Figure 2--for node 2 the times T_1 and T_2 can have any relation to each other, and T_2 may be less than T_3 (the time node 3 is entered). Again, T_1 (the time the Task Force detects the Recon) may be later than T_4 (the time the AEW aircraft detects the raid). Once beyond node 2, however, the timing is preserved.

C. Decision Variables

This section discusses decisions, decision variables, and their probabilities for both Red and Blue. A decision variable is essentially a controllable variable, it may or may not be in one-to-one correspondence with decision in the usual sense of the word. The term strategy as used in game theory is implicitly defined here in terms of a collection of decision variables. Indeed, the specification of a value for each decision variable constitutes a strategy[*].

1. Specification of Probabilities

Many of the important decision variables are binary (yes/no) and represent individual decisions. A convention has been adopted regarding notation: Blue's discrete decision variables are lower case b's, and Red's are lower case r's. Continuous variables use other notation. For example, the Blue variable b_1 is defined by:

$b_1 = 0$ if CAP will not be used

$b_1 = 1$ if CAP will be used.

This usage is for pure strategies only. More generally, we allow the "mixing" of this decision by introducing probabilities as decision variables:

$\text{Prob}(b_1 = 0)$ is the probability that CAP will not be used

$\text{Prob}(b_1 = 1)$ is the probability that CAP will be used.

? Alternatively and preferably, it is also possible to introduce b_1 as a component of a joint probability. For example, the decision as to whether the CV should be initially active (decision variable b_2) may be considered jointly with the CAP decision. If b_2 is defined to be zero for the CV

[*] Actually a behavioral strategy.

initially in EMCON and unity for the CV initially active, we may consider the four probabilities:

$$\text{Prob}(b_1 = 0,\ b_2 = 0)$$

$$\text{Prob}(b_1 = 0,\ b_2 = 1)$$

$$\text{Prob}(b_1 = 1,\ b_2 = 0)$$

$$\text{Prob}(b_1 = 1,\ b_2 = 1)$$

Previous page defines as decision variables b_1, b_2, r_1, r_2 etc. why the switch to Probabilities as decision variables.

as decision variables instead of $P(b_1 = 0)$, $P(b_1 = 1)$, $P(b_2 = 0)$, $P(b_2 = 1)$.

Choosing between these two formulations (independent specifications vs. joint specifications) is not simply a matter of taste. The two formulations are essentially those of behavioral strategies vs. adaptive strategies as discussed in Reference 1. It can be shown that the simply independent specification is often equivalent to the more complex joint specification. ? In any case, the joint specification is often ruled out from problem size considerations. For example, if there are ten decision variables with five levels each, the independent approach requires 10 × 5 = 50 probabilities and the joint approach requires 10 × 10 × 10 × 10 × 10 = 100,000 probabilities. For this research effort no hard and fast rule can be given for the form of probability specification. The principal guiding rule is one of pragmatism, joint specifications will be used to the extent that there is room for them, and when they are used it will be for those variables which most require joint specification.

How is this determined?

2. Node-by-Node Discussion

A node-by-node discussion of decisions and decision variables will now be undertaken. A symbol will be introduced for decision variables which are reasonably well-defined. We occasionally identify decisions and considerations from which further decisions could be defined in a more thorough study.

Node 1: Search

This phase will terminate in either No Detections or in one or more detections at times T_1, T_2 as shown on Figure 2. T_1 and T_2 are random variables selected from distributions in part determined by decision variables. They may be considered fixed for the remainder of the flow. Important Blue decisions at this node will continue to exert an influence over the problem flow in later states. Three such decisions are:

b_1: will Blue employ CAP and AEW?

b_2: will the CV be initially in EMCON, or active?

b_3: will the AEW aircraft search passively, or actively?

Note that b_3 is conditioned on b_2: if there are no AEW aircraft, then b_3 is irrelevant.

Red's major decision is also binary:

r_1: will Red search passively, or actively?

We are here implicitly considering a single search plan for the Red Recon, a more refined analysis might consider allowing Red to choose one of several search plans as an option. Another refinement requiring further Red decisions relates to the criterion for calling out the Red attack aircraft, should Red Recon do this on an ESM contact alone or should he wait for radar contact and more positive identification?

Node 2: Detection

Entry into this node implies that Blue has detected Red (at T_1), or Red has detected Blue (at T_2), or each has detected the other (at T_1, T_2, respectively). It is assumed that Red automatically calls up the attack aircraft at T_2, whatever its relation to T_1. Blue's major decision is:

b_4: will the CAP fighter attempt to intercept the Recon?

Two Red decisions are:

r_2: will the Recon retire from the scene following the communications to the attack aircraft?

r_3: will the Recon retire from the area when he discovers that he is being intercepted, even if he is needed by the incoming raid?

These three decisions would probably be based on consideratiOns not yet fully defined. From Blue's viewpoint, using the interceptor to attack the Recon reduces the defense capability against the Red raid (when it comes); besides, Blue is not sure of whether the Recon is actually required by the raid once initial CV position is determined and relayed. Red's decisions revolve around some of the same considerations: if he is not actually necessary he may linger around as a decoy, at some risk of being shot down. If Red Recon is necessary for the mission, he still will not choose to stay if he feels that he will be shot down with near certainty before the launch is consummated.

In any event, node 2 is exited to the node determined by the earliest of the following:

i) An intercept is planned for CAP vs. the Recon, with estimated intercept time T_3

ii) The raid is detected by AEW aircraft at T_4

iii) Attack Aircraft Launch their Missiles at T_5.

Node 3: Intercept is Planned for Time T_3 (CAP Intercepts Recon)

Entry into this node is at the time the planning decision took place, which may be taken to be time T_1 plus an appropriate delay. As indicated on Figure 2, the raid may or may not have already been detected by the AEW at time T_4.

This is an example of a node that is not a "game node," decisions already made together with various assumptions will determine the next node. Basically, there are three competing times: T_4, T_3, and T_5. If T_3 (the intercept) occurs first, the next node is Attack Aircraft Launch Missiles (node 9) if the intercept fails and Recon is Lost (node 5) if the intercept succeeds. If T_4 occurs before T_5 and both are before T_3, the transition is to Raid Detected By AEW at T_4 (node 4) or Attack Aircraft Launch Missiles, respectively.

Node 4: Raid Detected by AEW at T_4

It is understood that this node can be entered only when AEW aircraft are selected by Blue; this would be implemented by making transition probabilities on arcs into this node equal to zero. This node is an example of another kind of nongame node because Blue has decisions to make while Red does not. Instead of a game involving two-sided optimization by Blue and Red we have one-sided optimization by Blue alone.

Blue decides whether the CV should go active if it is now passive (b_5 = 1 means yes, 0 means no). Blue also decides whether to launch interceptors from the deck for an intercept at time T_8 (decision variable b_6), and whether or not to divert the CAP aircraft on intercept of the Recon to intercept the raid instead (decision variable b_7). Blue should consider these three binary decisions jointly (there are $2^3 = 8$ cases) and maximize the node 4 payoff function defined in terms of these decisions .

Node 5: Reconnaissance is Lost at T_3

The Recon is considered lost when it is either successfully intercepted or decides to leave the area, the latter being permitted by the earlier variables r_2 and r_3. In either case the time is T_3.

Both Blue and Red can be assumed to know of the occurrence of this event, and potentially there is an opportunity for making new decisions whose outcome depended upon the Recon being lost. For Red the problem is more critical if the Recon was actually necessary for success of the attack mission as we assume here. It has been assumed that the attackers fly at low altitude as long as possible, and loss of Recon implies loss of information on the CV.

The only decision defined for this node at present is Red decision r_4:

$r_4 = 0$ if the attackers return home without contact (node 12)

$r_4 = 1$ if the attackers elect to continue the mission, implying that one or more of them must pop-up to search for the CV.

Node 6: DLI Launched to Intercept Raid at T_6

The launch decision was made at node 4, intercept time is T_6. Node 6 may have been entered directly from node 4 or indirectly by the Recon is Lost-Attackers popup-CV detected popup path. The submodel at this node must determine transition probabilities for node 9 (Attack Aircraft Launch) and node 14 (Attack Aircraft Destroyed Before Launch). The transition out of node 6 depends on whether the T_5 is earlier than T_6. For particular times T_5 and T_6 this is simple to resolve. However, T_5 and T_6 are random variables, and the transition probability from node 6 to node 9 is actually $\text{Prob}(T_5 \leq T_6)$.

Nodes 7 and 8: Attackers Pop-up to Search for the CV and CV Detected Pop-up

These are combined and treated in detail in the next section, using decision variables D for Red and b_8 for Blue, where D = distance from planned launch point to pop-up point and $b_8 = 0$ if the CV remains in EMCON, $b_8 = 1$ if the CV goes active.

Node 9: Attack Aircraft Launch Missiles at T_5

This node is probably the most critical of all, for arrival in this node implies that the launch of ASCMs occurs. The transition may have been made from any of the nodes 2, 3, 4, 6, 7, or 8. Transitions out of node 9 are consistent with others on the diagram in their black-white character: either the missiles are detected by the CV or the missiles impact the CV.

Blue may be assumed to know of the launch if the raid had been detected by AEW aircraft. Blue decisions are dependent upon whether the launch was detected; if detected, Blue decides whether to divert CAP or DLI from any other intercept mission to the missiles (decision variable b_9) and whether to employ countermeasures such as chaff (decision variable b_{10}). Certain Red decisions (which would actually be made much earlier in the real world) may be considered made at this point in the model; in particular those controllable settings, thresholds, etc., which determine the missiles' flight profile and homing mode may be considered as node 9 decisions.

There will be uncertainty on both sides in a fully formulated game at this node. Blue will be uncertain as to whether the Red attack aircraft are needed by the missiles after launch. Red will be uncertain about those aspects of the problem that actually determine the optimal settings and thresholds but which are unknown at the time the settings are actually made.

Node 10: Missiles are Detected by CV

In this node Blue has decisions which relate to the use of chaff or other countermeasures, whether or not to divert CAP or DLI now on intercept mission, and whether to employ DLI against the missiles. These decisions have been mentioned before for earlier nodes, further analysis is needed to determine what node or nodes they should be in. Similarly for Red's decisions, the control settings may be more appropriate in this node than in node 9.

Nodes 11-15: Terminal Nodes

These are terminal nodes defined primarily for purposes of defining the payoff structure; there are no decisions made here.

V A DETAILED MODEL AT A GAME NODE

This section considers a particular node in some detail in order to demonstrate the methodology. Actually, a pair of nodes (7 and 8) are involved since it turns out to be analytically convenient to combine these two nodes into one which we will simply call node 7′. The new node 7′ then can be entered only from node 5 while transitions can be made to nodes 6, 9, or 12, according to Figure 2.

Node 7′ defines several states, one for each path from node 1 to node 7′ together with the times T_i associated with these paths. From this auxiliary information (path and times) one can construct a family of geometrical situations, an example of which is shown in Figure 4. Relative to some time reference not shown, T_1 and T_2 mark the respective positions of the Recon at times T_1 and T_2, while T_3 and T_4 mark the position of the raid at times T_3 and T_4, respectively. Missile launch will occur at time T_5 in the position shown if the attack aircraft are not shot down before reaching this point.

Also known in this node is a variable e , which makes its appearance for the first time. We may think of e as a variable initially set equal to the Blue decision variable b_2 (specifying will the CV be active initially?) and reset according to the other b_i which have to do with the later passive-to-active transition possibilities. Although e does not appear in Figure 2, it would be possible to introduce it there by adding nodes where the Blue decisions to go active can be made, and providing two branches out of these nodes corresponding to "remain passive" (e = 0) and "go active" (e = 1). The value of e will then be known because it is part of the path memory at a state.

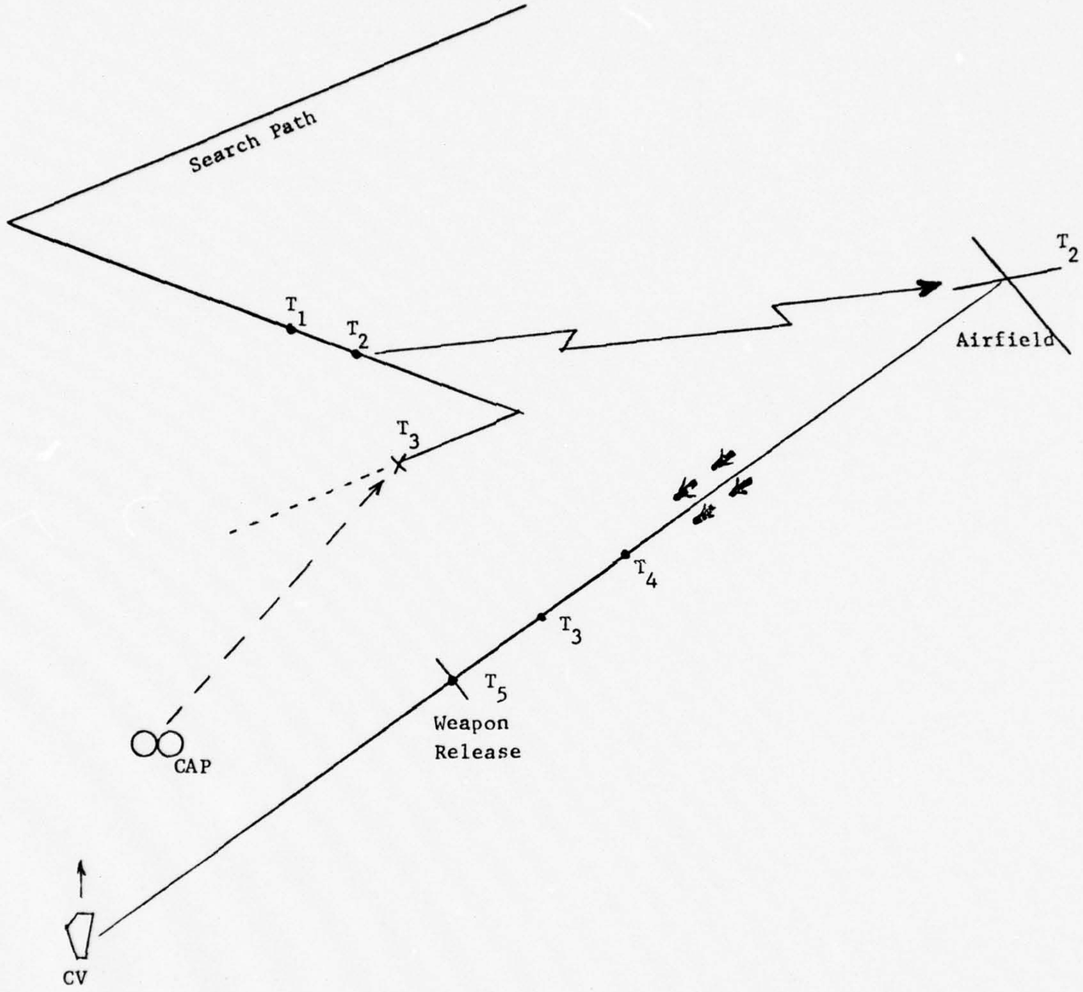


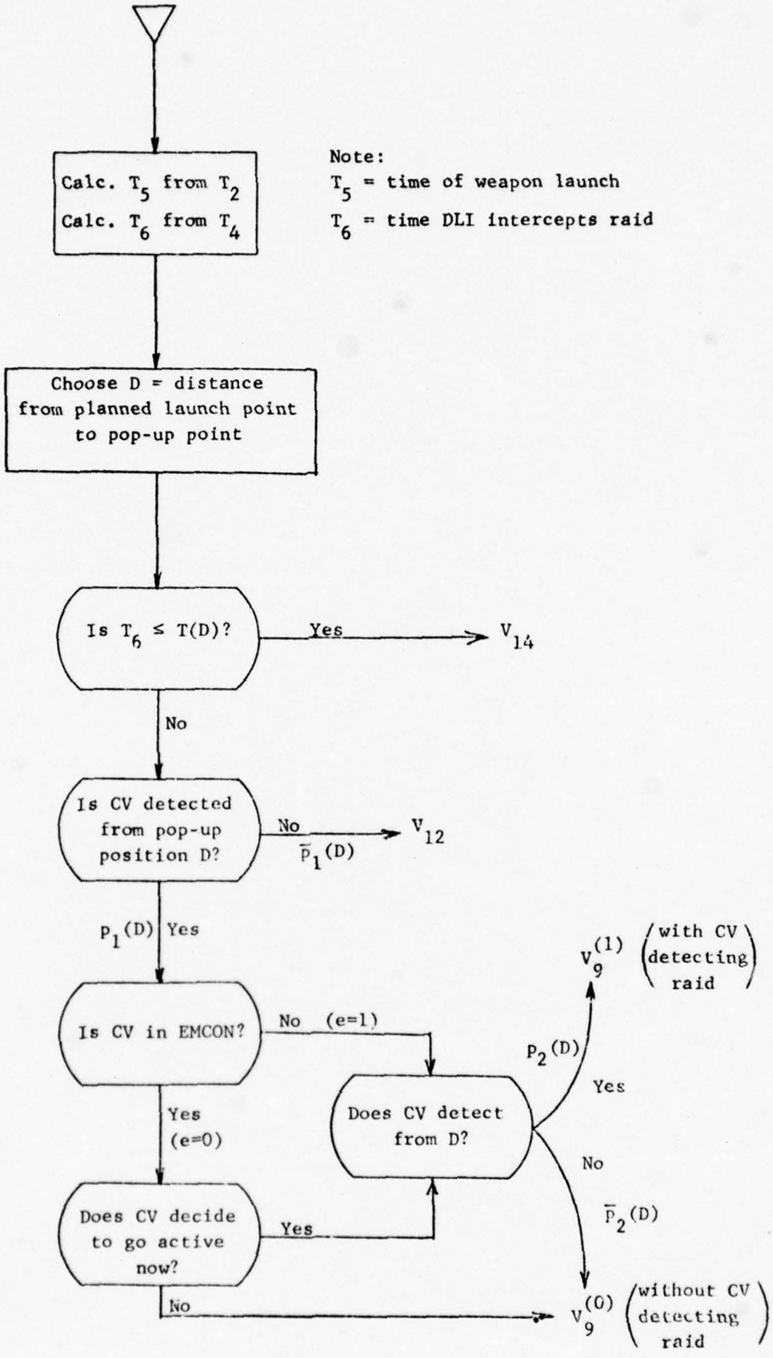
FIGURE 4   TYPICAL STARTING CONDITIONS FOR GAME AT NODE 7′

A. Monte Carlo Flow Diagram

Transition models associated with nodes can be expressed in many ways. It turns out to be convenient to define the transition model at node $7'$ in the form of a flow-chart for a simulation model. Equations will then be derived for payoff functions directly from the flow-chart, and the node $7'$ game then defined using these payoff functions to determine the payoff entries for an ordinary matrix game.

Figure 5 shows a flow-chart for the transition model of node $7'$, the aggregated node composed of nodes 7 and 8 in the original nodal diagram. The flow-chart is presented as though it were for a subroutine fitting into a larger computer program representing a Monte Carlo simulation of the entire effectiveness model. Program flow is therefore in the forward (increasing time) direction, the opposite of the dynamic programming algorithm. Inputs are shown at the top of the figure for the Monte Carlo subroutine interpretation, while the output for a given entry is the node to which flow is transferred at the exit. For convenience the payoff values (which are specified for terminal nodes and derived by dynamic programming for the internal nodes) are shown at the exits, as though these payoffs were "picked up" at the time of exit.

In outline, the flow is: find the time of weapon launch (T_5) and the time DLI intercepts the raid (T_6) from the detection times T_2 and T_4. These calculations will involve geometry from Figure 4 as well as the detection times. Because the decision has been made to pop-up before entering this node, Red must decide when to pop-up. This timing decision is defined implicitly in terms of a pop-up distance D, measured from the position of the previously planned launch point. The time corresponding to distance D is denoted by T(D), where $T(\cdot)$ is a known function. For a chosen D, the question "Is $T_6 \leq T(D)$?" means "Does the DLI intercept the raid before pop-up time?" If the answer is "yes,"



Note:
T_5 = time of weapon launch
T_6 = time DLI intercepts raid

FIGURE 5 FLOWCHART FOR STATE 7′ TRANSITION MODEL

the transition is to node 14 for payoff V_{14}, otherwise the flow continues with a detection check: "Is the CV detected by the attack aircraft popping up at distance D from the CV?" Absence of detection, which has probability $1 - p_1(D)$, implies another transition to terminal node 12 and payoff V_{12}. After deciding whether the CV has been detected, the other detection question is asked: "Does the CV detect the aircraft doing the pop-up?" For detection, it is assumed necessary that the CV must be active upon entry into node 7′ or go active because the Blue decision variable b_8 specifies it. When the CV is active, the probability of detection is assumed to be a known function $p_2(D)$. In a Monte Carlo sense the "yes" decision would be made a fraction $p_2(D)$ of the time, the "no" decision a fraction $(1 - p_2(D))$ of the time. In either case the transition is to node 9, with its derived payoffs $V_9(0)$ and $V_9(1)$ for the (no detection of raid by CV, detection of raid by CV) cases, respectively.

B. Solution of the Node Game

1. Exit Probabilities

The Monte Carlo model whose flow diagram is shown on Figure 5 need never be implemented as such. Instead, a Markov model equivalent to Figure 5 is used, from it the four probabilities of exit (to nodes 12, 14, and 9 with and without CV detection) determined analytically. It is not necessary to actually draw this equivalent diagram because the probabilities needed are already shown at the branch points in Figure 5. The general rule needed is: the probability of exiting at a node with payoff V_x is a sum over all paths from the entry point to node x. Each term of the sum is the product of the probabilities on branches along the path, i.e., the probability of the path.

?

Using this rule the probability of transferring to terminal node 12 ($p_{\cdot 12}$) is:

$$p_{\cdot 12} = P(T_8 > T(D))[1 - p_1(D)] \quad . \qquad (1)$$

Similarly, the probability of transferring to terminal node 14 ($p_{.14}$) is:

$$p_{.14} = P(T_6 \leq T(D)) \quad . \qquad (2)$$

Continuing with the two probabilities of transferring to node 9

$$P(\text{transfer to 9 without CV detecting raid}) \equiv p_{.90}$$

$$P(T_6 > T(D)) \cdot p_1(D) \times \left\{(1-e)[P(b_8 = 0) + P(b_8 = 1)\overline{p}_2(D)] + e\ \overline{p}_2(D)\right\} \qquad (3)$$

where a bar over a quantity signifies unity minus the quantity. Finally, since the four probabilities of transition out of the node must sum to one, the probability of transferring to node 9 with CV detection of the raid ($p_{.91}$) is:

$$p_{.91} = 1 - (p_{.12} + p_{.14} + p_{.90}) \quad . \qquad (4)$$

2. Payoff Function

Using formulas (1)-(4), the expected value of the payoff from node $7'$ onwards is:

$$V_{7'}(T_2,T_4,e;\ D,b_8) = p_{.12} \times V_{12} + p_{.14} \times V_{14} + p_{.90} \times V_9^{(0)} + p_{.91} \times V_9^{(1)} \quad . \qquad (5)$$

In equation (5), the $V_9^{(0)}$ and $V_9^{(1)}$ terms on the right are functions of T_2, T_4, and e just as the value term on the left. Equation (5) is the payoff function for this node. As it stands, $V_{7'}$ depends upon its five arguments T_2, T_4, e; D, and b_8. The last two (D and b_8) are decision variables which will be eliminated by the min-max operation.

3. Value Function

The value function at stage $7'$ results from playing a zero-sum, two-person game using the payoff function (5) to determine elements in the payoff matrix. The value function will be a function of the continuous variables T_2 and T_4 and the binary variable e . Since the game to be formulated in the next paragraph is an ordinary matrix game, it will not include T_2 and T_4 as explicit parameters in the answer. It will therefore be necessary to solve the node 7′ game at least eight times. (Two levels would be required for the binary variable e , and at least two levels each would be independently required by variables T_2 and T_4). Linear interpolation would be used in the variables T_2 and T_4, and higher order polynomials requiring additional game solutions would be used only if required.

4. Game Matrix

The payoff function in equation (5) can be written compactly as a function of D and of the probabilities of b_8 as shown in equation (6):

$$V_{7'} = f_1(D) + f_2(D) \cdot P(b_8 = 0) + f_3(D) \cdot P(b_8 = 1) \quad . \qquad (6)$$

This formula may be used to display the game matrix for a discrete game approximating the continuous one when D can take on only discrete levels D_1, D_2, ..., D_n. Table 1 shows the game matrix.

Table 1

GAME MATRIX FOR NODE 7′ GAME

	$P(D = D_1)$	$P(D = D_2)$	...	$P(D = D_n)$
$P(b_8 = 0)$	$f_2(D_1)$ $+ f_1(D_1)$	$f_2(D_2)$ $+ f_1(D_2)$	... ...	$f_2(D_n)$ $+ f_1(D_n)$
$P(b_8 = 1)$	$f_3(D_1)$ $+ f_1(D_1)$	$f_3(D_2)$ $+ f_1(D_2)$	... ...	$f_3(D_n)$ $+ f_1(D_n)$

The solution of the game in Table 1 yields a value $V_{7'}^{*}$, where the asterisk means "associated with the game solution." It is the value of the game that is played starting from node 7 in the original structure, i.e., it is the average payoff to Blue if both Red and Blue play optimal strategies from node 7 on. Optimal decision variables (often in the form of mixed strategies) are also game outputs which can be thought of as associated with the node; they are functions of the same variables as the value. In the Monte Carlo interpretation these variables would be used when the flow of the model arrived in node 7′. When needed, random numbers would be selected in accordance with the optimal probabilities and used to make the decisions which determine the next node.

VI MODELING UNCERTAINTY

One vital element is missing above, and that is the recognition and treatment of uncertainty in the real-world problem. Even in this relatively simple scenario there are several important places where uncertainty occurs:

i) The Recon does not know when or whether it has been detected by the AEW aircraft, or by the CV's radars

ii) The CV does not know when or whether it has been detected by the Recon

iii) The Recon does not know whether the CV has planned an attack on him, given detection.

It is sometimes difficult to know whether a particular matter which is uncertain in the ordinary sense of the word needs to be handled as uncertainty in the model. A way to proceed is to first define all the submodels for the several nodes in the game, assuming perfect information, and then examine them one at a time and ask: "Is the information assumed by the respective players actually available to each at this point?" When the information used by a party is in fact unknown, it then becomes a candidate for uncertainty modeling. The way we will proceed is by example, relegating the more general statement and model of uncertainty to Appendix B. Since node 7′ is the only node which has been described in detail, it is the node at which to look for uncertainty.

If one examines the earlier Figure 5, which shows the flow diagram for node 7′, one finds that it has been implicitly assumed that T_6, the time the DLI will intercept the raid, is known to Red as well as to Blue. Since T_6 is calculated from T_4 and geometry, this implies that Red knows T_4, the time that the AEW aircraft detects the raid. This is not a realistic assumption and we will now develop an uncertainty model to remedy the situation. Note the asymmetry here--Blue has perfect information[*] while Red has imperfect information (i.e., uncertainty). The discussion below applies to this situation only, while Appendix B is more general and treats the symmetrical case as well.

A. Probability Distribution Approach

It will turn out that a satisfactory theory can be developed if it is assumed that Red knows not T_4 itself but the probability distribution of T_4. Red will act as though the problem itself is random by playing into the probability distribution while Blue will respond in kind by incorporating into his own solution the knowledge of how Red will play. Thus Blue will use the same probability distribution in arriving at his own optimal solution. (Blue actually determines several solutions, conditioned on the true value of T_4. However, he uses only one of them, the one corresponding to the actual T_4, in a given play or Monte Carlo replication.)

The mathematics which incorporates this probability distribution and the assumptions associated with it is reasonably straightforward and will be presented shortly. Justifying the assumptions, or at least making them plausible, is a more difficult task than handling the mathematics. One way to try to do this is to consider another, quite different, "scenario."

Imagine that there are two analysts, a Blue analyst and a Red analyst, whose aim is to find optimal strategies for their respective military forces in the very scenario this document considers. They may realize early that a good deal of modeling will have to be done, and it may be possible to use each other's models. Indeed, going to the extreme in commonality, they may be able to use the same model providing uncertainty is properly treated.

[*] This is not technically correct without translating the structure into an equivalent game.

Suppose, for now, that Red uses an arbitrary estimate of the probability distribution of T_4. Suppose further that the gaming model has been completed and is ready to use. The analysts may run it for optimal decision variables (or strategies), then build a Monte Carlo version which employs these optimal strategies. They can then run the Monte Carlo model and derive various probability distributions--in particular, they can derive the probability distribution of T_4. This distribution could then replace the estimate Red started with, and the process repeated again and again until convergence.

Use deterministic model to get stopping solution? Then use S.S. in Monte Carlo model to simulate

→ doesn't this need a location of a probability distr. If so, what has been gained!

So far, this entire process appears to be just another iterative process of a type that arises frequently in applied Operations Research. The Blue and Red analysts have to now examine it and see if one is taking unfair advantage of the other. Since Red is controlling the distribution, it appears to be he who is somehow getting the better of the situation, if either is. What sort of complaint could Blue register about the process being biased against him?

With the possible exception of complications due to the introduction of uncertainty, Blue knows that they are involved in a zero-sum, two-person game. One of the more outstanding properties of the solutions to such games is that neither side can exploit the optimal strategies of the other, i.e., each side can give the other his own optimal strategy and lose nothing by it. Therefore, in this two-analyst context, the Blue analyst cannot lose anything by giving the Red analyst the optimal Blue strategy. (In fact, these strategies were already needed above to run the Monte Carlo model to obtain the distribution of concern.) Therefore, there seems to be no legitimate complaint Blue can raise about bias.

It would lead too far afield to pursue this matter further in this document. The critical point is that the postulated <u>common</u> model that was considered above really seems to exist--if shipboard facilities were available for Blue to actually put the model aboard ship he would have no incentive to change it after he and the Red analyst are finished with their joint venture. (A more formal related discussion is to be found in Reference 2 (Harsanyi). Harsanyi's context is n-person, non-zero sum games involving Nash equilibrium points and considerable game theory background is required.)

B. Mathematical Formulation

Turning to the mathematical formulation for the example, assume that $\{p_k\}$ is a discrete estimate of the probability distribution of T_4; specifically, let

$$p_k = \text{Prob}(T_4 = T_4^{(k)}) \qquad \text{for } k = 1,2,\ldots,K \quad , \tag{7}$$

where the $T_4^{(k)}$ are the approximating levels of T_4. Then, according to the model in Appendix B, the game matrix in Table 1 has to be duplicated K times and the kth duplicate scaled by p_k. These scaled duplicates are then stacked as shown in Table 2 to form a game matrix. The top block is Table 1 scaled by p_1, with the Blue decision variable b_8 superscripted by "1" to correspond to k = 1. The column structure (for Red's decision variables) is the same as it was in Table 1, reflecting a lack of choice for Red, while the duplication of blocks in the row structure reflects Blue's multiplicity of choices.

Denoting the entire game matrix in Table 2 by P, the Linear Programming tableau can be constructed (Table 3). In game theory parlance the problem has now assumed the form of a constrained matrix game. The upper left corner of the tableau is the payoff matrix P from Table 2. The row of ones below P expresses the probability constraint imposed on Red:

$$\sum_{j=1}^{n} P(D = D_j) = 1 \quad .$$

Table 2

GAME MATRIX FOR NODE 7′

	$P(D_1)$	$P(D_2)$	$P(D_n)$
$P(b_8^{(1)} = 0)$	$p_1(f_2(D_1) + f_1(D_1))$	$p_1(f_2(D_2) + f_1(D_2))$	$\ldots\ p_1(f_2(D_n) + f_1(D_n))$
$P(b_8^{(1)} = 1)$	$p_1(f_3(D_1) + f_1(D_1))$	$p_1(f_3(D_2) + f_1(D_2))$	$\ldots\ p_1(f_3(D_n) + f_1(D_n))$
$P(b_8^{(2)} = 0)$	$p_2(f_2(D_1) + f_1(D_1))$	...	
$P(b_8^{(2)} = 1)$	$p_2(f_3(D_1) + f_1(D_1))$	...	
	$\vdots$		
$P(b_8^{(k)} = 0)$	$p_k(f_2(D_1) + f_1(D_1))$	...	$p_k(f_2(D_n) + f_1(D_n))$
$P(b_8^{(k)} = 1)$	$p_k(f_3(D_1) + f_1(D_1))$	...	$p_k(f_3(D_n) + f_1(D_n))$

Table 3

LINEAR PROGRAMMING TABLEAU FOR TABLE 2

		Multipliers					
	D1 D2 ... ... Dn	λ_1	λ_2	λ_3	...	λ_k	
p_1 block	P	1 1					≤ 0
p_2 block			1 1				≤ 0
				1 1			≤ 0
					1 1		≤ 0
p_k block						1 1	≤ 0
μ_1	1 1 1 ... 1						= 1
Max Z:	0	1	1	1	1	1	←Obj Function

Corresponding to this row are several columns of ones expressing Blue's probability constraints:

$$P(b_8^{(k)} = 0) + P(b_8^{(k)}) = 1) = 1 \qquad \text{for } k = 1,2,\ldots,K \quad .$$

The bottom row expresses the objective function, which is to be maximized:

$$Z = \sum_{k=1}^{k} 1 \cdot \lambda_k \quad .$$

Defining β_j to be $P(D = D_j)$, the equation form of the primal (Red) problem is to choose $\beta_j \geq 0$ and any λ_k satisfying:

$$\sum_{j=1}^{n} p_{ij}\beta_j + \sum_{i=1}^{K} \lambda_{i'} \leq 0 \qquad i = 1,2,\ldots,2K \tag{8}$$

$$\sum_{j=1}^{n} \beta_j = 1 \tag{9}$$

to maximize

$$Z = \sum_{k=1}^{n} \lambda_k \tag{10}$$

where $P = (p_{ij})$ and $i' = i/2$ if i is even and $i' = (i+1)/2$ if i is odd.

The optimal primal solution to this constrained matrix game is a set of values for Red:

$$\beta_1^*,\ \beta_2^*,\ \ldots,\ \beta_n^*$$

and the multipliers

$$\lambda_1^*,\ \lambda_2^*,\ \ldots,\ \lambda_K^* \quad .$$

The optimal dual (Blue) solution comes as a by-product of the primal solution. Letting

$$\alpha_k = \text{Prob}(b_8^{(k)} = 0) \qquad \text{and} \qquad 1 - \alpha_k \equiv \overline{\alpha}_k = \text{Prob}(b_8^{(k)} = 1) \quad ,$$

the optimal Blue solution is:

$$\alpha_1^*,\ \alpha_2^*,\ \ldots,\ \alpha_k^* \qquad \text{and the multiplier } \mu_1^* \quad .$$

The value of the game is $- Z^*$, where Z^* is the optimal value of the objective function output by the linear program.

This solution is used in different ways, depending upon whether we are interested in the use in the game solution (by dynamic programming) or in the Monte Carlo interpretation of the model. In the Monte Carlo interpretation Red will use the mixed strategy dictated by the Prob $(D = D_j^*)$ whatever the true value of T_4 and Blue will use a mixed strategy for the true value (known to Blue, but not to Red) of T_4. (Interpolation may often be needed because in general the true T_4 will not equal one of the finite set $\{T_4^{(k)}\}$).

In the gaming solution the usage is different; instead of just one pair of values being used the entire solution is used. For the kth value of T_4 the value of the node 7′ game played when Blue has $T_4^{(k)}$ is:

$$V_{7'}^{*(k)} \equiv \sum_{j=1}^{n} \beta_j^* p_k \left[(f_2(D_j) + f_1(D_j))\alpha_k^* + (f_3(D_j) + f_1(D_j))(1 - \alpha_k^*)\right] \qquad (11)$$

This is simply the usual sum-of-double-products formula for the game value in terms of the strategies and the payoff matrix elements, modified by restricting the row player's variables to the kth block. Expressing this in terms of the earlier notation for value (test following Table 1) the game value at its arguments is:

$$V_{7'}^{*}(T_2, T_4^{(k)}, e) = V_{7'}^{(k)\,*}(T_2, e) \qquad (12)$$

where the dependence of the quantity on the right on T_2 and e has been made explicit.

Appendix A

A DETAILED DIAGRAM OF THE FLEET DEFENSE MODEL

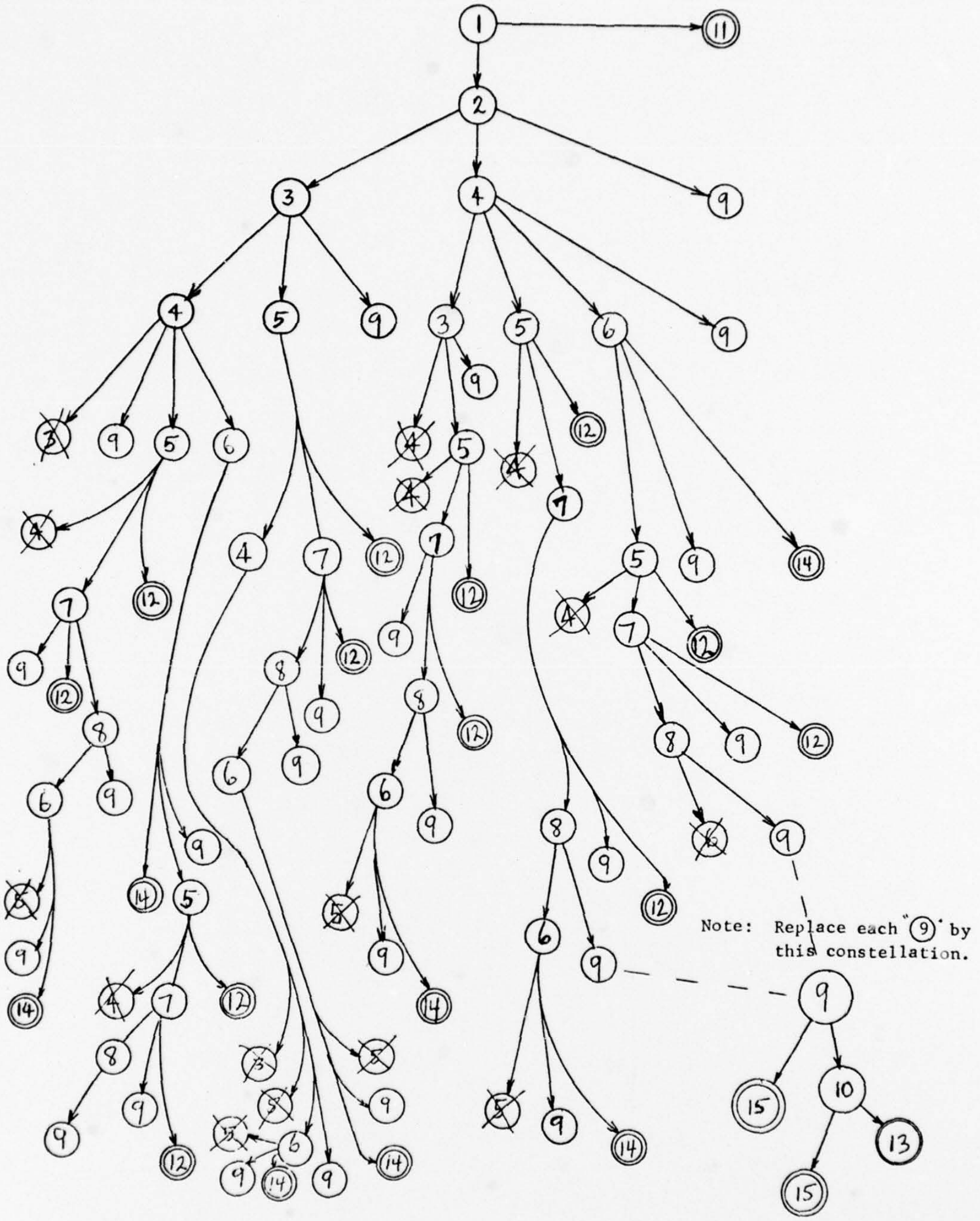
Appendix A

A DETAILED DIAGRAM OF THE FLEET DEFENSE MODEL

This appendix is devoted to further discussion of the Markov model derivable from Figure 2. Figure A-1 shows a tree derived from Figure 2 by tracing through the diagram from node to node, avoiding paths which enter the same node more than once. For clarity, only node numbers are shown. The main part of the graph is traced downwards to end in either a terminal node or in node 9, which is outside the loop. Node 9 and its successors are shown separately on the bottom of the page, this constellation would replace each node 9 in the main graph above if the graph were done in full.

Since the graph in Figure A-1 is a tree, there is a unique path from the circle representing node 1 to any other circle. Therefore, the circles can be identified with states as defined in the text. By simply counting the number of circles with a given node number, the number of states associated with each node in Figure 2 can be determined; the results are in Table A-1.

Because the length of a path is related in some sense to the complexity of the state, the distribution of path lengths is of some interest. Table A-2 shows the distribution derived from Figure A-1. Path length is defined to be the number of nodes appearing in its definition; e.g., path 1-2-3-5 has length four.



Note: Replace each "(9)" by this constellation.

FIGURE A-1 TREE DERIVED FROM EVENT-FLOW DIAGRAM

Table A-1

NUMBER OF STATES AT A NODE

Node Number	Number of States	
1	1	
2	1	
3	2	
4	4	
5	6	Total = 83
6	7	
7	6	
8	6	
9	25	
10	25	
11	1	
12	12	
13	25	Total = 96
14	8	
15	50	
	179	

Table A-2

PATH LENGTH DISTRIBUTION

Length of Path	Number of Paths
1	1
2	2
3	3
4	10
5	19
6	27
7	32
8	32
9	29
10	18
11	6
	179

Appendix B

A METHOD FOR INCORPORATING UNCERTAINTY WITH CONSTRAINED MATRIX GAMES

Appendix B

A METHOD FOR INCORPORATING UNCERTAINTY WITH CONSTRAINED MATRIX GAMES

This appendix contains a more general treatment of the uncertainty model introduced by example in the last section of the text. The theory is closely related to Harsanyi's more general model (References 2, 3, 4) which applies to n-person, non-zero sum games using Nash equilibrium points. (Two-person, zero-sum games are used as examples in the second of these references, and the theory appears to be almost identical to that expressed here. A difference is that Harsanyi uses the normal form of the game for solution, which implies joint strategies, while this appendix uses a constrained matrix game approach with independent strategies.) Harsanyi's set of papers require considerable background in game theory. An elementary example using the theory appears in Reference 5; a game in the Anti-Ballistic Missile field was formulated to incorporate uncertainty in the number of defensive interceptors.

The basic idea is simple: reduce the game with incomplete information to one with complete information by introducing a lottery which decides which of the several possible games will be played on a given trial. In the most general case, neither player knows the lottery outcome (i.e., knows the game being played). However, they do know the probability distribution governing the lottery and in addition they may have partial information about the lottery outcome.

We will proceed in steps rather than going directly to the most general symmetrical case. Suppose there are two subgames whose payoff matrices P_1 and P_2 are shown in Table B-1. Suppose further that a lottery will select game 1 with probability $p_1 = 1/3$ and game 2 with probability $p_2 = 2/3$.

As shown, the probability that Blue will choose column 1 is $x_1^{(g)}$ and $x_2^{(g)}$ ($= 1 - x_1^{(g)}$) is the probability that Blue will choose column 2 in game g . Similarly, y_1 and y_2 are Red's probabilities of selecting rows 1 and 2. For later reference, solutions of the games are shown when both Red and Blue know the game being played (game 1 has a pure strategy solution $x_2^{(1)} = y_2 = 1$ with value $\overline{V}_1^* = 0$. Game 2 played in the usual way has a mixed strategy solution $x_1^{(2)} = 3/8$, $x_2^{(2)} = 5/8$, $y_1 = 7/8$, $y_2 = 1/8$ with value $\overline{V}_2^* = 3/8$.

Now assume that Blue knows the outcome of the lottery, i.e., knows which game is being played, while Red knows only the probabilities p_1 and p_2.

Table B-1

PAYOFF MATRICES P_1 AND P_2 FOR TWO SUBGAMES

Game 1 ($p_1 = 1/3$)

	$x_1^{(1)} = 0$	$x_2^{(1)} = 1$
$y_1 = 0$	0	2
$y_2 = 1$	- 1	0^*

*(~ saddlepoint; value zero)

Game 2 ($p_2 = 2/3$)

	$x_1^{(2)} = 3/8$	$x_2^{(2)} = 5/8$
$y_1 = 7/8$	1	0
$y_2 = 1/8$	- 4	3

(Value 3/8)

Adopting Red's viewpoint, the average payoff $\overline{V}$ as a function of his probabilities y_1 and y_2 is found by conditioning on the subgame and summing:

$$\begin{aligned}\overline{V} &= \sum_{g=1}^{2} E(\overline{V}|\text{game} = g)\ P(\text{game} = g) \\ &= \left(y^T P_1 x^{(1)} \times \frac{1}{3}\right) + \left(y^T P_2 x^{(2)} \times \frac{2}{3}\right) \\ &= y^T\left(\frac{1}{3} P_1 x^{(1)} + \frac{2}{3} P_2 x^{(2)}\right) \quad . \qquad \text{(B-1)}\end{aligned}$$

Equation (B-1) is the payoff function for a new "lottery" game whose payoff matrix is shown in Table B-2.

Table B-2

PAYOFF MATRIX FOR LOTTERY GAME

	$x_1^{(1)}$	$x_2^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$
y_1	0	2/3	2/3	0
y_2	- 1/3	0	- 8/3	2

Optimal solutions* to the lottery game in Table B-2 are $x_1^{(1)*} = 0$, $x_2^{(1)*} = 1$, $x_1^{(2)*} = 1/4$, $x_2^{(2)*} = 3/4$ for Blue and $y_1^* = 7/8$, $y_2^* = 1/8$ for Red. The game value is $\overline{V}^* = 5/6$, which is larger than the average payoff to Blue from the two games played separately with perfect information. (That average is 1/3 × 0 + 2/3 × 3/8 = 1/4.) Therefore, as expected, Blue gains when Red has only probabilistic knowledge of the game being played.

*Found by a constrained matrix game.

Optimal strategies for the two situations may be compared to see what compromises have been made.

- In subgame 1 Blue uses the same pure strategy as in the perfect information (PI) game.
- In subgame 2 Blue shifts somewhat (from 3/8 probability to 1/4 probability for column 1).
- Red's optimal strategy is (coincidentally) identical to his optimal strategy for subgame 2, a reflection in part of the higher probability assigned to this game in the lottery.
- The "average PI strategy" for Red, defined to be the average of the perfect information strategies weighted by the lottery probabilities, works out to be $(\bar{y}_1, \bar{y}_2) = (7/12, 5/12)$, which is far from the optimal $y^* = (7/8, 1/8)$.

A more general and symmetrical formulation of a game model involving uncertainty is one in which each side has some one thing (such as a resource level) known to it and not to the other side. Let a_{ij} be the probability that the lottery will assign resource level i to Red and resource level j to Blue. Both Red and Blue will know the a_{ij} and for each given realization of the game Red will know i but not j and Blue will know j but not i. Let P_{ij} be the payoff matrix for the i,j case; both Blue and Red know the P_{ij} as well as the a_{ij}. (P_{ij} is the game matrix for the perfect information case in which Red has i, Blue has j, and both know i and j.)

The payoff matrix for this more general lottery game has a form which can be anticipated from the simpler case above. It is shown in Table B-3.

The $x^{(j)}$ are vectors of probabilities to be played if Blue's resource level is j. Similarly, the $y^{(i)}$ shown by the rows are vectors of probabilities to be played by Red. Individual elements $a_{ij}P_{ij}$ are submatrices of the total matrix, where the k,ℓth element of the submatrix

Table B-3

A MORE GENERAL LOTTERY GAME

	$x^{(1)}$	$x^{(2)}$		$x^{(n)}$
$y^{(1)}$	$a_{11} P_{11}$	$a_{12} P_{12}$	...	$a_{1n} P_{1n}$
$y^{(2)}$	$a_{21} P_{21}$	$a_{22} P_{22}$	...	$a_{2n} P_{2n}$
	$\vdots$			
$y^{(m)}$	$a_{m1} P_{m1}$	$a_{m2} P_{m2}$	...	$a_{mn} P_{mn}$

is a_{ij} times the k,ℓth element of P_{ij}. Such a payoff matrix will be used in a constrained matrix game formulation for solution by linear programming in the following example.

Example

A modification of the well-known Berkovitz Dresher N-stage game (Reference 6) illustrates the methodology for the symmetrical case. Consider a four stage game (N=4) in which the number of aircraft possessed by Red (Blue) is uncertain to Blue (Red) in the first stage only. (During the first stage, which may correspond to the start of a war, each side is assumed to be able to assess the other side's resources.) Uncertainty is assumed symmetrical; each side assumes the other has either 75%, 100%, or 125% of the actual number of aircraft. That is, if Red actually has 100 aircraft and Blue actually has 200 aircraft, then Blue will assume that Red has either 75, 100, or 125 aircraft, and Red will assume that Blue has 150, 200, or 250 aircraft. The three levels for each side are assumed equally likely, and independent. Therefore, the $3 \times 3 = 9$ values of a_{ij} all equal 1/9. Table B-4 summarizes the joint probability distribution (a_{ij}) and associated numbers of aircraft.

Table B-4

LOTTERY JOINT PROBABILITIES
AND INITIAL NUMBERS OF AIRCRAFT

		Blue		
		150	200	250
Red	75	1/9	1/9	1/9
	100	1/9	1/9	1/9
	125	1/9	1/9	1/9

In this model aircraft are assigned in some numbers to three missions (Counter-Air, Air Defense, and Ground Support) in such a way as to maximize (minimize) for Blue (Red) the difference between the total Blue ground support and the total Red ground support. Symbolically, let (x_i, u_i, m_i) be the number of Blue aircraft assigned to CA, AD, and GS missions, respectively, for stage i. Similarly, (y_i, w_i, n_i) are defined for Red. The payoff function is:

$$V = \sum_{i=1}^{4} (m_i - n_i) = \sum m_i - \sum n_i \quad .$$

p_i (and q_i) denote the number of Blue (and Red) aircraft at the beginning of stage i. The payoff function then becomes:

$$\overline{V} = \sum m_i - \sum n_i = \sum (p_i - x_i - u_i) - \sum (q_i - y_i - w_i)$$

because $p_i = x_i + u_i + m_i$ and $q_i = y_i + w_i + n_i$.

Attrition is calculated using the expressions:

$$p_{i-1} = (p_i - (y_i - u_i)^+)^+$$

$$q_{i-1} = (q_i - (x_i - w_i)^+)^+$$

where "+" means "positive part." Reading the p_{i-1} expression from the inside out, it says "first reduce the number of attackers y_i by one for each Blue defender, the result $(y_i - n_i)^+$ is the number of Red penetrators. Then reduce the number of Blue aircraft by one for each Red penetrator," the result is the number of Blue aircraft available at the beginning of stage i-1.

From the solution of the original problem it is known that for three stages the value of the game is (3 × number of Blue aircraft - 3 × number of Red aircraft). This is sufficient information to formulate a payoff matrix for four stages when the pure strategies for the fourth stage are given; a payoff entry is:

$$(p_4 - x_4 - u_4) - (q_4 - y_4 - w_4) + 3p_3 - 3p_3$$

where Blue chooses x_4, u_4, m_4 and Red chooses y_4, w_4, n_4.

The solution to be presented will be an approximation based on a small number of predetermined pure strategies for each side. Optimal strategies, which are not known, are probably not included in these sets*. Table B-5 shows Blue's pure strategies. The columns correspond to the Blue resource level assumed: 150, 200, or 250. There are eight rows and three columns in each matrix, corresponding to eight pure strategies for each of the three resource levels. For example, the first Blue strategy for the Blue resource level of 200 is $x_4 = 150$ (left table, first row and second column), $u_4 = 50$ (right table, first row and second column), with the excess going to Ground Support: $m_4 = 200-150-50 = 0$. Red's strategies were defined by similar table.

*However, the game value comes out to be quite close to the optimal, see final paragraph.

Table B-5

BLUE STRATEGIES

X Values

p_4 = 150	200	250
75	150	175
150	200	150
70	150	125
80	125	100
50	100	150
75	100	125
100	125	100
75	150	150

U Values

p_4 = 150	200	250
75	50	75
0	0	75
80	0	75
70	0	75
50	50	75
0	25	50
0	25	50
50	25	50

The game value for the perfect information solution to the $p_4 = 200$, $q_4 = 100$ case is:

$$V_4^* = 4.5(p_4 - q_4) = 450$$

as shown in Table II of Reference 6. Several values can be defined for the uncertainty game. The game value for the situation in which the resource levels are truly distributed according to the a_{ij} is 435, showing a slight advantage to the weaker player (Red). The uncertainty apparently hurts Red less than it hurts Blue. On the other hand, if the resource levels actually are 200 for Blue and 100 for Red the game value works out to be 402 (about 89% of the perfect information value) which favors Red even more.

Optimal strategies and their probabilities for this lottery game are given in Table B-6 as a function of actual number of aircraft. Probabilities for the lottery game are shown adjacent to the allocations. The last column on the right gives the optimal perfect information strategies from the original game when the resource levels are 200 for Blue, 100 for Red.

Table B-6

OPTIMAL STRATEGIES

Number of Aircraft	x_4	u_4	m_4	Lottery Game Probability	Perfect Information Game Probability
Blue					
150	80	70	0	0.682	
	100	0	50	0.318	
200	100	50	50	0.227	0
	150	50	0	0.318	1
	200	0	0	0.455	0
250	175	75	0	1.000	
Red	y_4	w_4	n_4		
75	75	0	0	0.182	
	0	75	0	0.409	
	0	0	75	0.409	
100	100	0	0	0.909	0.50
	0	100	0	0.091	0.50
125	0	125	0	1.0	

Some numerical work was done to partially bound the error due to the approximation of the strategy space. Red was allowed to optimize into the fixed Blue strategies (i.e., allocations and probabilities) shown in Table B-6. Red optimized separately for each of the three resource levels 75, 100, and 125. In all three cases the payoff could not be reduced below the game value, showing that one half of the approximation has zero error. The separate game values were:

$$V_{75}^{(R)} \equiv 534 \qquad \text{for} \quad Q = 75$$

$$V_{100}^{(R)} \equiv 445 \qquad \text{for} \quad Q = 100$$

$$V_{125}^{(R)} \equiv 326 \qquad \text{for} \quad Q = 125 \quad .$$

The average of these three values is the game value 435 mentioned earlier. The $V_Q^{(R)}$ terms are the game values when Blue has resource P = 150, 200, and 250 with equal probability, Red has Q, and both use strategies from Table B-6.

Allowing Blue to optimize similarly into Red resulted in:

$$V_{150}^{(B)} = 235$$

$$V_{200}^{(B)} = 435$$

$$V_{250}^{(B)} = 640$$

whose average is only 0.2* higher than the game value found by using the approximating strategies. In other words, the bounds on value provided by the two one-sided optimizations show that the game value is in error by only 0.2 in 435.

*Additional significant figures are needed to validate this.

REFERENCES

1. J. Bracken, J. E. Falk, and A. F. Karr, "Two Models for Optimal Allocation of Aircraft Sorties," Operations Research,Vol. 23, No. 5, (September-October 1975).

2. J. C. Harsanyi, "Games With Incomplete Information Played by "Bayesian" Players, Part I: The Basic Model," Management Science, Vol. 14, No. 3, (November 1967).

3. J. C. Harsanyi, "Games With Incomplete Information Played by "Bayesian" Players, Part II," Management Science, Vol. 14, No. 5, (January 1968).

4. J. C. Harsanyi, "Games With Incomplete Information Played by "Bayesian" Players, Part III: The Basic Probability Distribution of the Game," Management Science, Vol. 14, No. 7, (March 1968).

5. W. H. Frye and L. L. Ullman, "Methodology for Calculating Minuteman Defense Effectiveness," Technical Note SRD-EG14, Stanford Research Institute, Huntsville, Alabama (October 1973).

6. L. Berkovitz and M. Dresher, "A Game-Theory Analysis of Tactical Air War," Operations Research, Vol. 7, pp 599-620, (1959).